

# COMP3421

Introduction to 3D Graphics Solutions

# Exercises 1

Write a snippet of jogl code to draw a triangle with vertices:

$(2, 1, -4)$

$(0, -1, -3)$

$(-2, 1, -4)$

Make sure you specify face normals for the vertices.

# Exercises 2

We want to use a perspective camera to view our triangle. Which command/s would work?

`gl.glOrtho(-3,3,-3,3,0,8);`

`gl.glFrustum(-3,3,-3,3,0,8);`

`gl.glFrustum(-3,3,-3,3,-2,8);`

`glu.gluPerspective(60,1,2,8);`

`glu.gluPerspective(60,1,0,8);`

# Exercises 3

What would be an equivalent way to specify your perspective camera?

Where would the x and y vertices in our triangle be projected to on the near plane?

What would the pseudo-depth of our vertices be in CVV co-ordinates (-1..1)?

# Exercises 4

Suppose we wanted to add another triangle with vertices

$(-0.5, 0, 0)$

$(0.5, 0.5, 0)$

$(0.5, -0.5, 0)$

Would this appear on the screen? How could we fix this?

# Solution

See code for implementation

```
glu.gluPerspective(60,1,2,8);
```

```
top = near * tan(fov/2)
```

```
    = 2 * tan(30) = 1.155
```

```
bottom = -1.155
```

Since we have an aspect ratio of 1 the left and right would be the same,

```
gl.glFrustum(-1.155,1.155,-1.155,1.155,2,8)
```

# Solution

See code for implementation

Since we have not transformed our points, they are in camera co-ordinates. Our nearplane N is 2.  $x^* = NPx/-Pz$ ,  $y^* = NPy/-Pz$  ,

$$(2,1,-4) \quad x^* = 2*2/4 = 1 \quad y^* = 2/4$$

$$(0,-1,-3) \quad x^* = 2*0/3 = 0 \quad y^* = -2/3$$

$$(-2,1,-4) \quad x^* = 2*-2/4 = -1 \quad y^* = -2/4$$

# Solution

Pseudo-depth  $z^* = (aPz+b)/-Pz$

$a = (F+N)/F-N$ ,  $b = -2FN/(F-N)$

$a = (2+8)/(8-2) = 10/6$   $b = -2*2*8/(6)=-32/6$

$(2,1,-4) \ z^* = (-10/6*-4-32/6)/4 = 0.333$

$(0,-1,-3) \ z^* = (-10/6*-3-32/6)/3 = -0.333$

$(-2,1,-4) \ z^* = 0.333$



# Solution

We would have to move camera at least by 2 or more in the z direction. Here I have moved it by 2.5 (exactly 2 may not work as it may or may still get clipped).

```
gl.glTranslate(0,0,-2.5); OR
```

```
glu.gluLookAt(0,0,2.5,0,0,0,0,1,0);
```

We could not simply move the near plane to 0 as this is not valid for a perspective camera.